```
#include <stdio.h>
#include <stdlib.h>

#define n 100

// BFS traversal
void BFS(int adj_matrix[n][n], int vertices, int start)
{
   int visited[n] = {0};
   int queue[n];
   int front = -1, rear = -1;

   visited[start] = 1;
   queue[++rear] = start;

   printf("BFS Traversal: ");

   while (front != rear)
   {
      int current = queue[++front];
      printf("%d ", current);

      for (int i = 0; i < vertices; i++)
      {
```

```
            if (adj_matrix[current][i] && !visited[i])
            {
                visited[i] = 1;
                queue[++rear] = i;
            }
        }
    }
    printf("\n");
}

// DFS traversal
void DFS(int adj_matrix[n][n], int vertices, int start)
{
    int visited[n] = {0};
    int stack[n];
    int top = -1;

    visited[start] = 1;
    stack[++top] = start;

    printf("DFS Traversal: ");

    while (top != -1)
    {
```

```
        int current = stack[top--];
        printf("%d ", current);

        for (int i = 0; i < vertices; i++)
        {
            if (adj_matrix[current][i] && !visited[i])
            {
                visited[i] = 1;
                stack[++top] = i;
            }
        }
    }
    printf("\n");
}

int main()
{
    int vertices;
    printf("Enter the number of vertices: ");
    scanf("%d", &vertices);

    int adj_matrix[n][n];

    printf("Enter the adjacency matrix:\n");
```

```
    for (int i = 0; i < vertices; i++)
    {
        for (int j = 0; j < vertices; j++)
        {
            scanf("%d", &adj_matrix[i][j]);
        }
    }

    int start_vertex;
    printf("Enter the starting vertex for traversal: ");
    scanf("%d", &start_vertex);

    BFS(adj_matrix, vertices, start_vertex);

    DFS(adj_matrix, vertices, start_vertex);

    return 0;
}
```

```
Enter the number of vertices: 5
Enter the adjacency matrix:
0 1 1 0 0
1 0 0 1 1
1 0 0 0 0
0 1 0 0 0
0 1 0 0 0
Enter the starting vertex for traversal: 3
BFS Traversal: 3 1 0 4 2
DFS Traversal: 3 1 4 0 2
```